Low-Cost Inertial Measurement Unit Calibration With Nonlinear Scale Factors

Xin Zhang, Changle Zhou, Fei Chao, *Member, IEEE*, Chih-Min Lin, *Fellow, IEEE*, Longzhi Yang, *Senior Member, IEEE*, Changjing Shang, and Qiang Shen

***Abstract*—Inertial measurement units (IMUs) have been widely used to provide accurate location and movement measurement solutions, along with the advances of modern manufacturing technologies. The scale factors of accelerometers and gyroscopes are linear when the range of the sensors are reasonably small, but the factor becomes nonlinear when the range gets much bigger. Based on this observation, this article presents a calibration method for low-cost IMU by effectively deriving the nonlinear scale factors of the sensors. Two motion patterns of the sensor on a rigid object are moved to collect data for calibration: One motion pattern is to upcast and rotate the rigid object, and another pattern is to place the rigid object on a stable base in different attitudes. The rotation motion produces centripetal and Coriolis force, which increases the measurement range of accelerometers. Four cost functions with different weight factors and two sets of data are utilized to optimize the IMU parameters. The weight factor comes from derived formula with input values which are the variance of the noise of the sampled data. The proposed approach was validated and evaluated on both synthetic and real-world data sets, and the experimental results demonstrated the superiority of the proposed approach in improving the accuracy of IMU for long-range use. In particular, the errors of acceleration and angular velocity led by our algorithm are significantly smaller than those resulted from the existing approaches using the same testing data sets, demonstrating a remarkable improvement of 64.12% and 47.90%, respectively.**

***Index Terms*—Inertial measurement unit (IMU) calibration, low-cost IMU, nonlinear scale factors.**

Manuscript received February 8, 2021; revised April 14, 2021; accepted April 28, 2021. Date of publication May 4, 2021; date of current version October 27, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61673322 and Grant 61673326, in part by the Fundamental Research Funds for the Central Universities under Grant 20720190142, and in part by the Key Project of National Key R and D Project under Grant 2017YFC1703303. Paper no. TII-21-0595. *(Corresponding author: Fei Chao.)*

Xin Zhang and Changle Zhou are with the Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: shenshikexmu@qq.com; dozero@xmu.edu.cn).

Fei Chao is with the Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen 361005, China, and also with the Institute of Mathematics, Physics, and Computer Science, Aberystwyth University, SY23 3DB Aberystwyth, U.K. (e-mail: fchao@xmu.edu.cn).

Chih-Min Lin is with the Department of Electrical Engineering, Yuan Ze University, Tao-Yuan 32003, Taiwan (e-mail: cml@saturn.yzu.edu.tw).

Longzhi Yang is with the Department of Computer and Information Sciences, Northumbria University, NE2 1RW Newcastle upon Tyne, U.K. (e-mail: longzhi.yang@northumbria.ac.uk).

Changjing Shang and Qiang Shen are with the Institute of Mathematics, Physics, and Computer Science, Aberystwyth University, SY23 3DB Aberystwyth, U.K. (e-mail: cns@aber.ac.uk; qqs@aber.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3077296.

Digital Object Identifier 10.1109/TII.2021.3077296

I. Introduction

An INERTIAL measurement unit (IMU) is often used to detect the acceleration and angular velocity of a moving object. Such measurements provide important navigation information (i.e., position, velocity, attitude, etc.) in aerospace or other space exploration applications [1]. Thanks to the advances of microelectrophoretical techniques, inertial sensors can be manufactured as microelectrophoretical system (MEMS), with considerably low cost but with increased accuracy, thereby greatly improving the applicability of IMU in many new areas, such as body activity classification [2], sports analysis [3], and indoor localization [4]. However, MEMS inertial sensors still sometimes suffer from systematic and stochastic errors [5]. Systematic errors are caused by poorly calibrated system dynamics; thus, these errors can be eliminated in theory by involving very careful calibration. In contrast, stochastic errors are regarded as white noise with zero mean that comes from measurement error; hence, such errors cannot be eliminated. IMU calibration is, therefore, a critical process in an effort to reduce the systematic errors.

Systematic errors are practically handled by a set of adjustable parameters, including inaccurate scaling, sensor axis misalignments, cross-axis sensitivities, and nonzero biases. These parameters are set as vectors or matrices in most of the implementations of IMU calibration, [6], [7]. With the help of these parameters, through matrix multiplication operations, the original data of a sensor with systematic errors can be transformed into more accurate data without systematic errors. A number of optimization algorithms, such as "least square method" [6] and "Kalman filter" [7], [8], are often employed to optimize these parameters. Because of the limited size of MEMS sensor devices, the scale factor of an accelerator is a nonlinear parameter. For instance, if the value along the x-axis is 1 g, the scale factor is 0.9; however, if the value is 5 g, the scale factor changes to 1.1. If the value is $-6\,g$, the scale factor is 1.2. The scale factor of the gyroscope device is also nonlinear. Therefore, the scale factor must be

nonlinearly represented in order to better reduce the systematic errors.

Studies have attempted to nonlinearly model the scale factor of accelerometers. A method was proposed for the calculation of a nonlinear parameter for the scale factor of an accelerometer [9]; however, because the data for calibration is collected only in the gravitational field, the calculated scale factor of each axis is limited to the range of [–1g, 1g]. Such range-limited parameters, thus, cannot be used for the tasks with large gravity value. To overcome this, the work reported in [10] suggested rotating a sensor around a point to detect centripetal and Euler acceleration so as to expand the detection range of IMU accelerometer [11]–[13]. This article used a turntable platform to create the rotation and then to detect the accelerometer parameters, and the Kalman filter is used to obtain nonlinear parameters. However, the radius of rotation is calculated from a linear scale factor, which affects the accuracy of the scale factor parameters. Therefore, the specification of the values of a radius vector remains a challenge.

This article proposes a method for obtaining nonlinear scale factors of a sensor without using high-cost instruments to address the aforementioned challenge. In this method, the IMU is affixed to the surface of an object that cannot be easily deformed. To acquire the raw data of the IMU sensor, two types of actions are required using the object: 1) rotating the object and then placing it on a stable base at different attitudes, 2) throwing and rotating the object along different axes of rotation; then letting the object to fall freely and catching it by hand. A filter algorithm in this approach is designed to extract useful information from the raw data. With these data, all initial parameters of the IMU sensor are placed into four cost functions. Weighted least squares are adopted in this work as the cost functions to optimize the parameters. Each cost function weight is derived from the inverse of the variance of the cost function, which is obtained by accumulating the variance of the noise from the sampled data. The proposed work has been validated and evaluated by experiments. The main contributions of this article are summarized as follows.

1) A method is proposed to obtain IMU calibration parameters that include the nonlinear scale factor. The method covers a paradigm of data collection and algorithm for calculating parameters.
2) The weight factors of the cost functions are derived according to the variance of the noise of the sampled data.

The rest of this article is organized as follows. Section II presents the background of the IMU error model and introduces the four cost functions for the IMU calibration and global cost function composed by these four cost functions. Section III describes the calibration, static detector, *Runge–Kutta* integration and cumulative variance, and weight parameters of cost function. Section IV reports the experimental results of the proposed algorithm on synthetic and real data. Finally, Section V concludes this article.

II. IMU Error Model and Calibration Cost Function

Biases, scale factor errors, cross-coupling errors, and random noise exist in accelerometers and gyroscopes. Higher order errors and angular rate-acceleration cross-sensitivity may also present [14]. Nonlinearity errors in the scale factor determine that the scale factor varies with specific force or angular velocity. To compensate these errors, the calibration model of the accelerometer is defined as follows:

$$\boldsymbol{a}^o + \delta\boldsymbol{a}^o = \boldsymbol{T}^a * \boldsymbol{f}^a(\boldsymbol{a}^s + \delta\boldsymbol{a}^s + \boldsymbol{b}^a, \boldsymbol{K}^a). \tag{1}$$

The input of this equation are the raw data of accelerometers, which are artificially divided into two parts, true states $\boldsymbol{a}^s = [a_x^s, a_y^s, a_z^s]^T$, and the error states $\delta\boldsymbol{a}^s = [\delta a_x^s, \delta a_y^s, \delta a_z^s]^T$. The true states $\boldsymbol{a}^s$ are the ideal measurement values in sensor frames. The error states $\delta\boldsymbol{a}^s$ are the measurement errors that cannot be eliminated and are independent from $\boldsymbol{a}^s$. Each triad of $\delta\boldsymbol{a}^s$ is an independent white noise with a mean value of zero. The vector, $\boldsymbol{b}^a = [b_x^a, b_y^a, b_z^a]^T$, represents the bias compensation. The scale factor function $\boldsymbol{f}^a$ produces a 3×1 dimensional vector of 3-order nonlinear equations for each axis

$$\boldsymbol{f^a}\left(\begin{bmatrix}\boldsymbol{a}_x\\ \boldsymbol{a}_y\\ \boldsymbol{a}_z\end{bmatrix}, \boldsymbol{K^a}\right) = \begin{bmatrix}k_{11}^a\boldsymbol{a}_x + k_{12}^a\boldsymbol{a}_x^2 + k_{13}^a\boldsymbol{a}_x^3\\ k_{21}^a\boldsymbol{a}_y + k_{22}^a\boldsymbol{a}_y^2 + k_{23}^a\boldsymbol{a}_y^3\\ k_{31}^a\boldsymbol{a}_z + k_{32}^a\boldsymbol{a}_z^2 + k_{33}^a\boldsymbol{a}_z^3\end{bmatrix} \tag{2}$$

where $\boldsymbol{K}^a = [k_{11}^a, k_{12}^a, k_{13}^a, k_{21}^a, k_{22}^a, k_{23}^a, k_{31}^a, k_{32}^a, k_{33}^a]$ represents the parameters of $\boldsymbol{f}^a$. The reason for choosing the 3-order polynomial function is that the nonlinear system requires an odd function and an even function. The theoretical basis of this consideration comes from any continuous function that can be expressed as a combination of an odd function and an even function. The quadratic function is a simple even function; in addition, the cubic function is a simple odd function. Therefore, the scale factor function $\boldsymbol{f}$ is produced as a 3-order polynomial nonlinear equation.

$\boldsymbol{T}^a$ is a 3×3 matrix that eliminates the effects of the cross-coupling errors and converts the data from nonorthogonal to orthogonal. Conventionally, inspired by [15], for small angles, a measurement $\boldsymbol{a}^S$ in a nonorthogonal frame can be transformed in the orthogonal body frame $\boldsymbol{a}^B$

$$\boldsymbol{a}^B = \boldsymbol{T}\boldsymbol{a}^S, \boldsymbol{T} = \begin{bmatrix}1 & -\theta_{yz} & \theta_{zy}\\ \theta_{xz} & 1 & -\theta_{zx}\\ -\theta_{xy} & \theta_{yx} & 1\end{bmatrix}. \tag{3}$$

In the presented calibration method, we employ the accelerometers orthogonal frame AOF [6], where θ_{xz}, θ_{xy}, and θ_{yx} become zero. Therefore, in the case of accelerometers, $\boldsymbol{T}^a$ becomes

$$\boldsymbol{T}^a = \begin{bmatrix}1 & -\alpha_{yz} & \alpha_{zy}\\ 0 & 1 & -\alpha_{zx}\\ 0 & 0 & 1\end{bmatrix}. \tag{4}$$

The calibrated outputs of interferometers are also divided into two parts: true states, $\boldsymbol{a}^o = [a_x^o, a_y^o, a_z^o]^T$ transformed from $\boldsymbol{a}^s$, and error states, $\delta\boldsymbol{a}^o = [\delta a_x^o, \delta a_y^o, \delta a_z^o]^T$ transformed from $\delta\boldsymbol{a}^s$. The mean of $\delta\boldsymbol{a}^o$ is $\boldsymbol{0}_{3\times1}$ and $\delta\boldsymbol{a}^o$ cannot be eliminated. The superscript $\boldsymbol{s}$ denotes the raw data from sensor; the superscript $\boldsymbol{o}$ represents the data is in calibrated frame; the superscript $\boldsymbol{a}$ represents the acceleration parameters.

Similarly, the calibration model of the gyroscope is defined as

$$\boldsymbol{\omega}^o + \delta\boldsymbol{\omega}^o = \boldsymbol{T}^g * \boldsymbol{f}^g(\boldsymbol{\omega}^s + \delta\boldsymbol{\omega}^s + \boldsymbol{b}^g + \boldsymbol{G}^g(\boldsymbol{a}^o + \delta\boldsymbol{a}^o), \boldsymbol{K}^g). \tag{5}$$

The inputs are the true state $\boldsymbol{\omega}^s = [\omega_x^s, \omega_y^s, \omega_z^s]^T$ and the error states $\delta\boldsymbol{\omega}^s = [\delta\omega_x^s, \delta\omega_y^s, \delta\omega_z^s]^T$ in the sensor frames. The gyroscope bias compensation is $\boldsymbol{b}^g = [b_x^g, b_y^g, b_z^g]^T$. An additional parameter g-dependent bias $\boldsymbol{G}^g$ [14] is defined as

$$\boldsymbol{G}^g = \begin{bmatrix} G_1 & G_2 & G_3 \\ G_4 & G_5 & G_6 \\ G_7 & G_8 & G_9 \end{bmatrix}. \tag{6}$$

The low-cost IMU gyroscopes measure the Coriolis force [14] to obtain the angular velocity. The specific force $\boldsymbol{a}^o$ also affects the oscillating mass. Therefore, $\boldsymbol{G}^g$ (i.e., the g-dependent bias matrix) is employed here to remove the effect of specific force. The scale factor function $\boldsymbol{f}^g$ produces 3×1 dimensional vector for each axis

$$\boldsymbol{f^g}\left(\begin{bmatrix} \boldsymbol{\omega}_x \\ \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z \end{bmatrix}, \boldsymbol{K^g}\right) = \begin{bmatrix} k_{11}^g \boldsymbol{\omega}_x + k_{12}^g \boldsymbol{\omega}_x^2 + k_{13}^g \boldsymbol{\omega}_x^3 \\ k_{21}^g \boldsymbol{\omega}_y + k_{22}^g \boldsymbol{\omega}_y^2 + k_{23}^g \boldsymbol{\omega}_y^3 \\ k_{31}^g \boldsymbol{\omega}_z + k_{32}^g \boldsymbol{\omega}_z^2 + k_{33}^g \boldsymbol{\omega}_z^3 \end{bmatrix} \tag{7}$$

where $\boldsymbol{K}^g = [k_{11}^g, k_{12}^g, k_{13}^g, k_{21}^g, k_{22}^g, k_{23}^g, k_{31}^g, k_{32}^g, k_{33}^g]$ represents the parameters of $\boldsymbol{f}^g$; and superscript $\boldsymbol{g}$ represents the gyroscope parameters. $\boldsymbol{T}^g$ guarantees that the gyroscope and accelerometer measurement refers to the same frame in the AOF case [6]. Therefore, in the gyroscope case, $\boldsymbol{T}^g$ becomes

$$\boldsymbol{T}^g = \begin{bmatrix} 1 & -\gamma_{yz} & \gamma_{zy} \\ \gamma_{xz} & 1 & -\gamma_{zx} \\ -\gamma_{xy} & \gamma_{yx} & 1 \end{bmatrix}. \tag{8}$$

The outputs are the true states, $\boldsymbol{\omega}^o = [\omega_x^o, \omega_y^o, \omega_z^o]^T$, and error states, $\delta\boldsymbol{\omega}^o = [\delta\omega_x^o, \delta\omega_y^o, \delta\omega_z^o]^T$, in the calibrated frames; in addition, the coordinate systems of ω^o are identical with that of $\boldsymbol{a}^o$.

A. Cost Function of Accelerometer

The parameters in the matrix of the accelerometers can be represented as a vector: $\boldsymbol{\theta}^a = [\alpha_{yz}, \alpha_{zy}, \alpha_{zx}\ , b_x^a, b_y^a, b_z^a, k_{11}^a, k_{12}^a, k_{13}^a, k_{21}^a, k_{22}^a, k_{23}^a, k_{31}^a, k_{32}^a, k_{33}^a]$. For simplicity, the calibration model of the accelerometer, as defined in (1), can be restated as follows:

$$\boldsymbol{a}^o + \delta\boldsymbol{a}^o = \boldsymbol{h}(\boldsymbol{a}^s, \delta\boldsymbol{a}^s, \boldsymbol{\theta}^a) = \boldsymbol{T}^a * \boldsymbol{f}(\boldsymbol{a}^s + \delta\boldsymbol{a}^s + \boldsymbol{b}^a, \boldsymbol{K}^a). \tag{9}$$

If the sensor is in a stationary state, the resultant output force of the corrected accelerometers is $1g$. Thus, the cost function of the accelerometer calibration can be defined as follows:

$$\boldsymbol{L}_a(\boldsymbol{\theta}^a) = \sum_{k=1}^{M} (\|\boldsymbol{h}(\boldsymbol{a}_k^s, \boldsymbol{\theta}^a)\|^2 - \|\boldsymbol{g}\|^2)^2 \tag{10}$$

where the vectors $\boldsymbol{a}_k^s$ measured in the sensor frame are the means of acceleration data in a temporal window t_w; and the acceleration data is collected from the sensor in different (M) attitudes.

B. Cost Function of Gyroscopes

The unknown parameters for gyroscope calibration can be represented as: $\boldsymbol{\theta}^g = [\gamma_{yz}, \gamma_{zy}, \gamma_{xz}, \gamma_{zx}, \gamma_{xy}, \gamma_{yx}, b_x^g, b_y^g, b_z^g, k_{11}^g, k_{12}^g, k_{13}^g, k_{21}^g, k_{22}^g, k_{23}^g, k_{31}^g, k_{32}^g, k_{33}^g, G_1, G_2, G_3, G_4, G_5, G_6, G_7, G_8, G_9]$. Two cost functions are employed for the gyroscope calibration in two different states. When the sensor is in a stationary state, the ideal output of the gyroscope is the vector $\boldsymbol{0}_{3\times1}$ which can be defined as

$$\boldsymbol{L}_g^1(\boldsymbol{\theta}^g) = \sum_{k=1}^{M} \|\boldsymbol{\omega}_k^s + \boldsymbol{b}^g + \boldsymbol{G}^g \boldsymbol{a}_k^o\|^2 \tag{11}$$

where $\boldsymbol{\omega}_k^s$ are the means of the gyroscope data in a temporal window t_w, and $\boldsymbol{\omega}_k^s$ and $\boldsymbol{a}_k^s$ are measured simultaneously. $\boldsymbol{a}_k^o$ is calibrated from $\boldsymbol{a}_k^s$ by following (9).

When the sensor is moving, the gyroscopes record changes in the attitudes as the gravity vector changes along with the attitude of the sensor. This means the accelerometers detect different gravity vectors at different attitudes in the moving state, which can be concisely represented as

$$\boldsymbol{a}_k^o + \delta\boldsymbol{a}_k^o = (\boldsymbol{R}_{k-1,k} + \delta\boldsymbol{R}_{k-1,k})(\boldsymbol{a}_{k-1}^o + \delta\boldsymbol{a}_{k-1}^o) \tag{12}$$

where the initial gravity vector is $\boldsymbol{a}_{k-1}^o$, and $\delta\boldsymbol{a}_{k-1}^o$ is the measurement errors of $\boldsymbol{a}_{k-1}^o$; $\boldsymbol{a}_k^o$ is the gravity vector after the change of attitude, and $\delta\boldsymbol{a}_k^o$ is the measurement errors of $\boldsymbol{a}_k^o$; $\boldsymbol{R}_{k-1,k}$ represents the rotation matrix, indicating the transformation from the $k-1$ attitude to the k attitude, and $\delta\boldsymbol{R}_{k-1,k}$ is the measurement errors of $\boldsymbol{R}_{k-1,k}$ that come from the error in angular velocity measurement. Rotation matrix $\boldsymbol{R}_{k-1,k}$ is derived from the quaternion $\boldsymbol{q} = [q_0, q_1, q_2, q_3]^T$ using the following:

$$\boldsymbol{R} = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & 1-2q_1^2-2q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1-2q_1^2-2q_2^2 \end{bmatrix}. \tag{13}$$

The quaternion $\boldsymbol{q}$ represents the attitude. The updating of quaternion $\boldsymbol{q}_n$ is achieved from the accumulation of the sequence of n angular velocity $\boldsymbol{\omega}_i^o$. Therefore, the cost function for the gyroscope calibration in moving state is defended as

$$\boldsymbol{L}_g^2(\boldsymbol{\theta}^g) = \sum_{k=2}^{M} \|\boldsymbol{a}_k^o - \boldsymbol{R}_{k-1,k}\boldsymbol{a}_{k-1}^o\|^2. \tag{14}$$

C. Cost Function of Weightless IMU

As the three axes of the accelerometer are capsuled in a $3 \times 3 \times 1$ mm^3 cubic space, for simplicity, it is presumed that the sensitive axes of the accelerometer triad meet at one point in this article. The error led by this presumption is neglectable in comparison to the errors that arise from nonlinearity [16]. IMU is always attached to a rigid ball when utilized.

When the rigid ball rotates, the acceleration $\boldsymbol{a}_s$ detected in the sensor frame can be defined as [14]

$$\boldsymbol{a}_s = \boldsymbol{a}_b + \boldsymbol{\omega}_b \times \boldsymbol{\omega}_b \times \boldsymbol{r}_{bs} + 2\boldsymbol{\omega}_b \times \dot{\boldsymbol{r}}_{bs} + \dot{\boldsymbol{\omega}}_{\boldsymbol{b}} \times \boldsymbol{r}_{bs} + \ddot{\boldsymbol{r}}_{bs} \tag{15}$$

where $\boldsymbol{a}_b$ denotes the inertial acceleration of the rigid body; $\boldsymbol{r}_{bs}$ is the radius from the center of gravity to the origin of sensor frame; $\dot{\boldsymbol{r}}_{bs}$ and $\ddot{\boldsymbol{r}}_{bs}$ are the first and second derivatives of $\boldsymbol{r}_{bs}$,

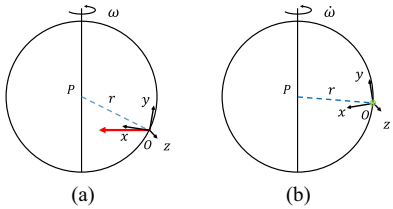

Fig. 1. (a) Centrifugal force in IMU. Red arrow represents centripetal force, which is in a plane with angular velocity $\boldsymbol{\omega}$ and radius $\boldsymbol{r}$ simultaneously vertical to angular velocity $\boldsymbol{\omega}$. (b) Euler acceleration in IMU. Green point represents Euler acceleration, which is an arrow vertical to angular velocity $\boldsymbol{\omega}$ and radius $\boldsymbol{r}$ and point to paper inside.

respectively; $\boldsymbol{\omega}_b$ is the vector of angular velocity in the rigid body frame; $\dot{\boldsymbol{\omega}}_b$ is the first derivative of $\boldsymbol{\omega}_b$. The term $\boldsymbol{\omega}_b \times \boldsymbol{\omega}_b \times \boldsymbol{r}_{bs}$ represents a centripetal acceleration. The term $2\boldsymbol{\omega}_b \times \dot{\boldsymbol{r}}_{bs}$ is known as Coriolis acceleration [14] and the term $\dot{\boldsymbol{\omega}}_b \times \boldsymbol{r}_{bs}$ is an Euler acceleration (or tangential acceleration).

The term $\boldsymbol{a}_b$ is the combined force of all the forces received by the rigid body. $\boldsymbol{a}_b$ includes gravity and air resistance of the rigid body during vertical movement. Gravity accelerates the rigid ball downward. If ignoring the air resistance, the vertical force felt by the sensor is $\boldsymbol{0}_{3\times1}$. Since the air resistance in the direction of motion is related to the velocity of the rigid body, it is necessary to reduce the impact of air resistance. If the rigid body is thrown at the highest point, the vertical speed is 0, and the air resistance is also 0. Controlling the height of the tossed rigid ball will effectively reduce the vertical movement speed of the rigid ball. The radius of the rigid ball is 0.1 m. If the height of the rigid body tossed is 0.1 m, the maximum speed is 1.40 m/s, and the maximum air resistance is 0.012 N.

As the rigid body $\boldsymbol{r}_{bs}$ is a constant, $\dot{\boldsymbol{r}}_{bs}$ and $\ddot{\boldsymbol{r}}_{bs}$ are $\boldsymbol{0}_{3\times1}$. The Coriolis acceleration $2\boldsymbol{\omega}_b \times \dot{\boldsymbol{r}}_{bs}$ is $\boldsymbol{0}_{3\times1}$. When the rigid body is at the highest point after tossed, the sensor can only detect centripetal acceleration and Euler acceleration

$$\boldsymbol{a}_s = \boldsymbol{\omega}_s \times \boldsymbol{\omega}_s \times \boldsymbol{r}_{bs} + \dot{\boldsymbol{\omega}}_s \times \boldsymbol{r}_{bs}. \tag{16}$$

1) Centripetal Acceleration: The centrifugal force of IMU is illustrated in Fig. 1(a), where $\boldsymbol{\omega}$ represents the angular velocity; $\boldsymbol{P}$ is the center of gravity of the rigid body; $\boldsymbol{O}$ is the origin of sensor frame. The radius $\boldsymbol{r}$ is the vector from $\boldsymbol{O}$ to $\boldsymbol{P}$. The centripetal acceleration $\boldsymbol{f}_C(\boldsymbol{\omega}, \boldsymbol{r})$ is calculated as

$$\boldsymbol{f}_C(\boldsymbol{\omega}, \boldsymbol{r}) = \begin{bmatrix} (\omega_y^2 + \omega_z^2)r_x - \omega_x\omega_y r_y - \omega_x\omega_z r_z \\ -\omega_x\omega_y r_x + (\omega_x^2 + \omega_z^2)r_y - \omega_y\omega_z r_z \\ -\omega_x\omega_z r_x - \omega_y\omega_z r_y + (\omega_x^2 + \omega_y^2)r_z \end{bmatrix} \tag{17}$$

where $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$ and $\boldsymbol{r} = [r_x, r_y, r_z]^T$.

2) Euler Acceleration: If a subject throws a rigid body to rotate in the air, the resultant force from the hand does not completely act on the rigid center of gravity. The resultant force causes the rigid body's rotation axis to precess around a certain axis. The change in the rotation axis produces the Euler acceleration. Due to the conservation of energy, the combined speed of rotation is constant so that the combined speed will be used in the determination of weightlessness of rigid bodies.

The Euler acceleration is illustrated in Fig. 1(b), where $\dot{\boldsymbol{\omega}}$ is the first derivative of $\boldsymbol{\omega}$. Note that the directions of $\boldsymbol{\omega}$ and $\dot{\boldsymbol{\omega}}$ are not in parallel in most cases; then the Euler acceleration is calculated as follows:

$$\boldsymbol{f}_E(\dot{\boldsymbol{\omega}}, \boldsymbol{r}) = \begin{bmatrix} \dot{\omega}_z r_y - \dot{\omega}_y r_z \\ \dot{\omega}_x r_z - \dot{\omega}_z r_x \\ \dot{\omega}_y r_x - \dot{\omega}_x r_y \end{bmatrix} \tag{18}$$

where $\dot{\boldsymbol{\omega}} = [\dot{\omega}_x, \dot{\omega}_y, \dot{\omega}_z]^T$. Using (17) and (18) to rewrite (16), the cost function of the IMU in weightless situation is then defined as follows:

$$\boldsymbol{L}_{\text{weightless}}(\boldsymbol{\theta}^a, \boldsymbol{\theta}^g, \boldsymbol{r}) = \sum_{j=1}^{N} \|\boldsymbol{a}_j^o - \boldsymbol{f}_C(\boldsymbol{\omega}_j^o, \boldsymbol{r}) - \boldsymbol{f}_E(\dot{\boldsymbol{\omega}}_j^o, \boldsymbol{r})\|^2 \tag{19}$$

where N is the number of frames of data in the weightless state detected by the static detector operator; $\dot{\boldsymbol{\omega}}$ in the term $\boldsymbol{f}_E(\dot{\boldsymbol{\omega}}, \boldsymbol{r})$ can be calculated using

$$\dot{\boldsymbol{\omega}}_j = \frac{\boldsymbol{\omega}_{j-2}}{12\Delta t} - \frac{2\boldsymbol{\omega}_{j-1}}{3\Delta t} + \frac{2\boldsymbol{\omega}_{j+1}}{3\Delta t} - \frac{\boldsymbol{\omega}_{j+2}}{12\Delta t} + O(\Delta t^4) \tag{20}$$

where $\boldsymbol{\omega}_{j-2}, \boldsymbol{\omega}_{j-1}, \boldsymbol{\omega}_j, \boldsymbol{\omega}_{j+1}, \boldsymbol{\omega}_{j+2}$ are samples of the gyroscope in interval Δt.

D. Global Cost Function

The weights of the cost functions are used to constitute a global cost function, which can be defined as

$$\begin{aligned} \boldsymbol{L}(\boldsymbol{\theta}^a, \boldsymbol{\theta}^g, \boldsymbol{r}) = &\sum_{k=1}^{M} \boldsymbol{C}_1(\|\boldsymbol{h}(\boldsymbol{a}_k^s, \boldsymbol{\theta}^a)\|^2 - \|\boldsymbol{g}\|^2)^2 \\ &+ \sum_{k=1}^{M} \boldsymbol{C}_2\|\boldsymbol{\omega}_k^s + \boldsymbol{b}^g + \boldsymbol{G}^g\boldsymbol{a}_k^o\|^2 \\ &+ \sum_{k=2}^{M} \boldsymbol{C}_3\|\boldsymbol{a}_k^o - \boldsymbol{R}_{k-1,k}\boldsymbol{a}_{k-1}^o\|^2 \\ &+ \sum_{j=1}^{N} \boldsymbol{C}_4\|\boldsymbol{a}_j^o - \boldsymbol{f}_C(\boldsymbol{\omega}_j^o, \boldsymbol{r}) - \boldsymbol{f}_E(\dot{\boldsymbol{\omega}}_j^o, \boldsymbol{r})\|^2 \end{aligned} \tag{21}$$

where the weight coefficients $\boldsymbol{C}_1, \boldsymbol{C}_2, \boldsymbol{C}_3$, and $\boldsymbol{C}_4$ enable the cost functions to have the same contribution to the global cost function during parameter optimization. The weight coefficients are the reciprocals of the variances of the cost functions, which will be detailed in Section III-C. The following section introduces the procedure of the calibration algorithm.

III. Calibration Procedure

The proposed calibration protocol is shown in Fig. 2, which consists of a data collection part and a parameter calculation part. In the **data collection** stage, two types of raw IMU data (RD1 and RD2) must be collected from different sensor motions. For RD1, the sensor is set at M different poses. Each pose is stably placed for at least t_w s. In addition, the time of the first pose placed is set to T_{init} s, longer than t_w. The transition from a pose to another pose is manually controlled. Thus, data of the sensor in stationary and movement can be then acquired.

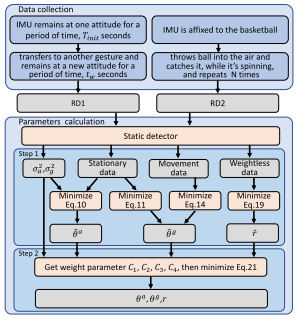

Fig. 2. Diagram of calibration protocol.

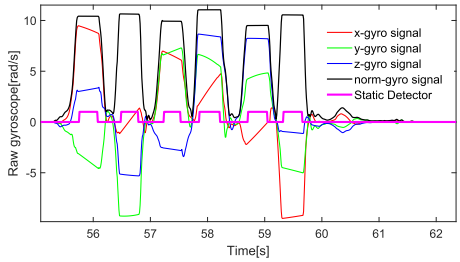

Fig. 3. Example of static detector for RD2. The red line represents the static detector.

The **parameter calculation** is performed in two steps: **Step 1** initializes the parameters $\boldsymbol{\theta}^a$, $\boldsymbol{\theta}^g$, and $\boldsymbol{r}$. Through filtering, the raw data RD1 is divided into stationary data and movement data; the raw data RD2 is for the presumption of weightless IMU (denoted as "weightless data"). The stationary data are fed into the cost functions as specified in (10) and (11), with $\widehat{\boldsymbol{T}^a}$, $\widehat{\boldsymbol{K}^a}$, $\widehat{\boldsymbol{b}^a}$, $\widehat{\boldsymbol{b}^g}$, and $\widehat{\boldsymbol{G}^g}$ being the output. The movement data and $\widehat{\boldsymbol{T}^a}$, $\widehat{\boldsymbol{K}^a}$, $\widehat{\boldsymbol{b}^a}$, $\widehat{\boldsymbol{b}^g}$, $\widehat{\boldsymbol{G}^g}$ are substituted into the cost function defined in (14), leading to the pre-estimated parameters $\widehat{\boldsymbol{T}^g}$ and $\widehat{\boldsymbol{K}^g}$. Note that, for simplicity, only the first columns k_{11}, k_{21}, k_{31} of $\widehat{\boldsymbol{K}^a}$ and $\widehat{\boldsymbol{K}^g}$ are calculated; the other items are set as 0. Then, the weightless data and $\widehat{\boldsymbol{T}^a}, \widehat{\boldsymbol{K}^a}, \widehat{\boldsymbol{b}^a}, \widehat{\boldsymbol{b}^g}, \widehat{\boldsymbol{G}^g}\widehat{\boldsymbol{T}^g}, \widehat{\boldsymbol{K}^g}$ are used to calculate the radius $\hat{\boldsymbol{r}}$ from the center of the sensor to the center of the rigid object. **Step 2** optimize the initialized parameters set in Step 1 in an effort to optimize the global cost function as defined in (21).

The *Levenberg–Marquardt (LM)* algorithm is employed here to minimize these cost functions as demonstrated by (10), (11), (14), (19), and (21). After several iterations, if the difference between the cost function of the current parameters and the cost function of the previous step parameters is less than a preset threshold, then the parameters have reached the optimal values. Note that the *LM* algorithm is merely used as an optimization tool, which solves the optimization problem of 45 parameters in the cost function. It is difficult to simultaneously optimize 45 parameters for general problems. If the initial values are not selected properly, it is easy to fall into local optimums, which makes the calibration failed. The calibration algorithm proposed in this work might fall into the local optimum only in setting the initial value.

For the convergence issue, first of all, IMU has an absolute calibration parameter, in the case of excluding random walks. Second, the absolute calibration parameter has the minimum value under each subcost functions (10), (11), (14), (19), and global cost function (21). In the calculation of the initial value, if the initial parameters $\hat{\theta}^a$, $\hat{\theta}^g$, and $\hat{r}^a$ satisfy the minimum subcost function value, the estimation is close to the absolute calibration parameter. Therefore, substituting these initial values into the global cost function can obtain a closer estimation of the absolute calibration parameters.

A. Static Detector

As discussed earlier, the streams of raw data RD1 and RD2 are divided into three forms: stationary, movement, and weightless data. In the static state, although a bias is existing, the resultant velocity of rotation does not change. Conversely, the gyroscope's value will change during the movement. In the weightless state, the resultant velocity of rotation is constant.

The variance-based static detector operator as reported in the work of [17] is used here as a shear. From this, the variance of the gyroscopes, $[\omega_x^t; \omega_y^t; \omega_z^t]$ is defined as

$$\varsigma(t) = \mathrm{var}_{t_w}\left(\sqrt{(\omega_x^t)^2 + (\omega_y^t)^2 + (\omega_z^t)^2}\right) \tag{22}$$

where $\mathrm{var}_{t_w}()$ is an operator to compute the variance in a time window, t_w s, and t represents a time point which is the center of the time window.

A threshold is defined as the multiplication of the square of the variance magnitude ς_{init} from (22) and $\boldsymbol{n}$ for all data captured in the initialization period of T_{init}. The length of T_{init} is calculated by *Allan* variance [18]. To classify data, a simple check is built to determine whether the value of $\varsigma(t)$ is less or greater than the threshold $\boldsymbol{n}\varsigma_{\text{init}}$. The result of the static detector is separated into M segments of stationary data and M–1 segments of movement data. Fig. 3 shows the segment of three-axis angular velocity in RD2. The static detector is shown by a red line, which indicates that the sensor is in a weightless state.

B. Attitude Integration and Cumulative Variance

The attitude of the sensor is expressed in the form of a quaternion $\boldsymbol{q}$. The differential of $\boldsymbol{q}$ describes the quaternion kinematics, which can be expressed as [19]

$$\begin{bmatrix} q_0^{k+1} \\ q_1^{k+1} \\ q_2^{k+1} \\ q_3^{k+1} \end{bmatrix} = \begin{bmatrix} \cos\frac{\theta}{2} & -\frac{\theta_x}{\theta}\sin\frac{\theta}{2} & -\frac{\theta_y}{\theta}\sin\frac{\theta}{2} & -\frac{\theta_z}{\theta}\sin\frac{\theta}{2} \\ \frac{\theta_x}{\theta}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} & \frac{\theta_z}{\theta}\sin\frac{\theta}{2} & -\frac{\theta_y}{\theta}\sin\frac{\theta}{2} \\ \frac{\theta_y}{\theta}\sin\frac{\theta}{2} & -\frac{\theta_z}{\theta}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} & \frac{\theta_x}{\theta}\sin\frac{\theta}{2} \\ \frac{\theta_z}{\theta}\sin\frac{\theta}{2} & \frac{\theta_y}{\theta}\sin\frac{\theta}{2} & -\frac{\theta_x}{\theta}\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \begin{bmatrix} q_0^k \\ q_1^k \\ q_2^k \\ q_3^k \end{bmatrix} \tag{23}$$

where $[\theta_x, \theta_y, \theta_z] = [\omega_x, \omega_y, \omega_z]\Delta t$ is three-axes angular in unit time from three-axes angular velocity, and $\theta = \sqrt{{\theta_x}^2 + {\theta_y}^2 + {\theta_z}^2}$.

In the presence of noise $\delta\boldsymbol{\omega}$ in angular velocity measurement, the attitude update of IMU is similar to Gaussian process. Then calculate the variance of the accumulated error, assuming that the variance of the cumulative altitude error of the kth time is $D[\boldsymbol{q}^k + \delta\boldsymbol{q}^k] = [\sigma^2_{q_0^k}, \sigma^2_{q_1^k}, \sigma^2_{q_2^k}, \sigma^2_{q_2^k}]^T$. The derivation of the

cumulative altitude error of the $k+1$th time is as follows:

$$\begin{aligned}
&\boldsymbol{D}\begin{bmatrix} q_0^{k+1}+\delta q_0^{k+1} \\ q_1^{k+1}+\delta q_1^{k+1} \\ q_2^{k+1}+\delta q_2^{k+1} \\ q_3^{k+1}+\delta q_3^{k+1} \end{bmatrix} \\
&\approx \boldsymbol{D}\begin{bmatrix} \begin{bmatrix} 1 & -\frac{\theta_x+\delta\theta_x}{2} & -\frac{\theta_y+\delta\theta_y}{2} & -\frac{\theta_z+\delta\theta_z}{2} \\ \frac{\theta_x+\delta\theta_x}{2} & 1 & \frac{\theta_z+\delta\theta_z}{2} & -\frac{\theta_y+\delta\theta_y}{2} \\ \frac{\theta_y+\delta\theta_y}{2} & -\frac{\theta_z+\delta\theta_z}{2} & 1 & \frac{\theta_x+\delta\theta_x}{2} \\ \frac{\theta_z+\delta\theta_z}{2} & \frac{\theta_y+\delta\theta_y}{2} & -\frac{\theta_x+\delta\theta_x}{2} & 1 \end{bmatrix} \begin{bmatrix} q_0^k+\delta q_0^k \\ q_1^k+\delta q_1^k \\ q_2^k+\delta q_2^k \\ q_3^k+\delta q_3^k \end{bmatrix} \end{bmatrix} \\
&\approx \begin{bmatrix} \sigma^2_{q_0^k}+\frac{\Delta t^2}{4}((q_1^k)^2\sigma^2_{\omega_x}+(q_2^k)^2\sigma^2_{\omega_y}+(q_3^k)^2\sigma^2_{\omega_z}) \\ \sigma^2_{q_1^k}+\frac{\Delta t^2}{4}((q_0^k)^2\sigma^2_{\omega_x}+(q_2^k)^2\sigma^2_{\omega_z}+(q_3^k)^2\sigma^2_{\omega_y}) \\ \sigma^2_{q_2^k}+\frac{\Delta t^2}{4}((q_0^k)^2\sigma^2_{\omega_y}+(q_1^k)^2\sigma^2_{\omega_z}+(q_3^k)^2\sigma^2_{\omega_x}) \\ \sigma^2_{q_3^k}+\frac{\Delta t^2}{4}((q_0^k)^2\sigma^2_{\omega_z}+(q_1^k)^2\sigma^2_{\omega_y}+(q_2^k)^2\sigma^2_{\omega_x}) \\ +\frac{\Delta t^2}{4}(\omega_x^2\sigma^2_{q_1^k}+\omega_y^2\sigma^2_{q_2^k}+\omega_z^2\sigma^2_{q_3^k}) \\ +\frac{\Delta t^2}{4}(\omega_x^2\sigma^2_{q_0^k}+\omega_z^2\sigma^2_{q_2^k}+\omega_y^2\sigma^2_{q_3^k}) \\ +\frac{\Delta t^2}{4}(\omega_y^2\sigma^2_{q_0^k}+\omega_z^2\sigma^2_{q_1^k}+\omega_x^2\sigma^2_{q_3^k}) \\ +\frac{\Delta t^2}{4}(\omega_z^2\sigma^2_{q_0^k}+\omega_y^2\sigma^2_{q_1^k}+\omega_x^2\sigma^2_{q_2^k}) \end{bmatrix}.
\end{aligned} \tag{24}$$

C. Weight Parameters of Cost Function

The weight coefficients $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, $\boldsymbol{C}_3$, and $\boldsymbol{C}_4$ are derived from the reciprocals of the variances of the cost functions. When the calibration parameters are accurate, the variance of the cost functions is originated from the white noise of the measurement data. The variances $[\sigma^2_{a_x^s},\sigma^2_{a_y^s},\sigma^2_{a_z^s}]^T$ and $[\sigma^2_{\omega_x^s},\sigma^2_{\omega_y^2},\sigma^2_{\omega_z^s}]^T$ in the sensor frame are obtained by the statistics of RD1 at the beginning T_{init} interval. The variance $\boldsymbol{\sigma}^2_{a^o}$ of $\delta\boldsymbol{a}^o$ in calibrated frame is $[\sigma^2_{a_x^o},\sigma^2_{a_y^o},\sigma^2_{a_z^o}]^T = [{k_{11}^a}^2\sigma^2_{a_x^s},{k_{21}^a}^2\sigma^2_{a_y^s},{k_{31}^a}^2\sigma^2_{a_z^s}]^T$, whilst the variance $\boldsymbol{\sigma}^2_{\omega^o}$ of $\delta\boldsymbol{\omega}^o$ is $[\sigma^2_{\omega_x^o},\sigma^2_{\omega_x^o},\sigma^2_{\omega_z^o}]^T = [{k_{11}^g}^2\sigma^2_{\omega_x^s},{k_{21}^g}^2\sigma^2_{\omega_y^s},{k_{31}^g}^2\sigma^2_{\omega_z^s}]^T$. The parameters such as k_{11}^a and k_{11}^g are obtained from the pre-estimation of $\widehat{\boldsymbol{\theta}}^a$ and $\widehat{\boldsymbol{\theta}}^g$. The effects of $\boldsymbol{T}^a$ and $\boldsymbol{T}^g$ are ignored when calculating the calibrated frame variances. For the convenience of calculation and expression, the three components of $\boldsymbol{\sigma}^2_{a^o}$ are replaced by their mean values $\bar{\sigma}^2_{a^o} = \frac{\sigma^2_{a_x^o}+\sigma^2_{a_y^o}+\sigma^2_{a_z^o}}{3}$, $\bar{\sigma}^2_{\omega^s} = \frac{\sigma^2_{\omega_x^s}+\sigma^2_{\omega_y^s}+\sigma^2_{\omega_z^s}}{3}$, and $\bar{\sigma}^2_{\omega^o} = \frac{\sigma^2_{\omega_x^o}+\sigma^2_{\omega_y^o}+\sigma^2_{\omega_z^o}}{3}$, respectively.

The variance of the cost function as specified in (10) is

$$\begin{aligned}
&\boldsymbol{D}[\|\boldsymbol{h}(\boldsymbol{a}^s,\delta\boldsymbol{a}^s,\boldsymbol{\theta}^a)\|^2-\|\boldsymbol{g}\|^2] \\
&\quad = 2(\sigma^4_{a_x^o}+\sigma^4_{a_y^o}+\sigma^4_{a_z^o})+4[{a_x^o}^2\sigma^2_{a_x^o}+{a_y^o}^2\sigma^2_{a_y^o}+{a_z^o}^2\sigma^2_{a_z^o}].
\end{aligned} \tag{25}$$

The variance of the cost function (11) is

$$\begin{aligned}
&\boldsymbol{D}[\boldsymbol{\omega}_k^s+\delta\boldsymbol{\omega}_k^s+\boldsymbol{b}^g+\boldsymbol{G}^g(\boldsymbol{a}_k^o+\delta\boldsymbol{a}_k^o)] \\
&\quad = \begin{bmatrix} \sigma^2_{\omega_x^s}+{G_1}^2\sigma^2_{a_x^o}+{G_2}^2\sigma^2_{a_y^o}+{G_3}^2\sigma^2_{a_z^o} \\ \sigma^2_{\omega_y^s}+{G_4}^2\sigma^2_{a_x^o}+{G_5}^2\sigma^2_{a_y^o}+{G_6}^2\sigma^2_{a_z^o} \\ \sigma^2_{\omega_z^s}+{G_7}^2\sigma^2_{a_x^o}+{G_8}^2\sigma^2_{a_y^o}+{G_9}^2\sigma^2_{a_z^o} \end{bmatrix}.
\end{aligned} \tag{26}$$

The variance of the cost function (14) can be expressed as

$$\begin{aligned}
&\boldsymbol{D}[\boldsymbol{a}_k^o+\delta\boldsymbol{a}_k^o-(\boldsymbol{R}_{k-1,k}+\delta\boldsymbol{R}_{k-1,k})(\boldsymbol{a}_{k-1}^o+\delta\boldsymbol{a}_{k-1}^o)] \\
&\quad = \begin{bmatrix} \boldsymbol{g}^2(4\sigma^2_{q^m}+8\sigma^4_{q^m})+2\bar{\sigma}^2_{a^o} \\ \boldsymbol{g}^2(4\sigma^2_{q^m}+8\sigma^4_{q^m})+2\bar{\sigma}^2_{a^o} \\ \boldsymbol{g}^2(4\sigma^2_{q^m}+8\sigma^4_{q^m})+2\bar{\sigma}^2_{a^o} \end{bmatrix}.
\end{aligned} \tag{27}$$

In this equation, the cumulative error $\sigma^2_{q^m}$ is led by the angular velocity error. For the convenience of calculation, the variance of each component of $\boldsymbol{q}^m$ is set as a common value, and the variance iteration formula of a single component can be expressed as

$$\begin{cases} \sigma^2_{q^0}=0 \\ \sigma^2_{q^1}\approx\frac{\Delta t^2}{4}\bar{\sigma}^2_{\omega^o} \\ \sigma^2_{q^{i+1}}\approx(1+\frac{3\Delta t^2}{4}\bar{\boldsymbol{\omega}}^2)\sigma^2_{q^i}+\frac{\Delta t^2}{4}\bar{\sigma}^2_{\omega^o} \end{cases} \tag{28}$$

where $\bar{\boldsymbol{\omega}}$ is the average of the absolute value of the angular velocity in the movement data.

The variance of the cost function (19) is

$$\begin{aligned}
&\boldsymbol{D}[\boldsymbol{a}^o+\delta\boldsymbol{a}^o-\boldsymbol{f}_C(\boldsymbol{\omega}^o+\delta\boldsymbol{\omega}^o,\boldsymbol{r})-\boldsymbol{f}_E(\dot{\boldsymbol{\omega}}^o+\delta\dot{\boldsymbol{\omega}}^o,\boldsymbol{r})] \\
&\quad = \begin{bmatrix} \boldsymbol{\omega}^2_{\max}\bar{\sigma}^2_{\omega^o}[(2\widehat{r}_x-\widehat{r}_y)^2+(2\widehat{r}_x-\widehat{r}_z)^2+(\widehat{r}_y+\widehat{r}_z)^2] \\ \boldsymbol{\omega}^2_{\max}\bar{\sigma}^2_{\omega^o}[(2\widehat{r}_y-\widehat{r}_x)^2+(2\widehat{r}_y-\widehat{r}_z)^2+(\widehat{r}_x+\widehat{r}_z)^2] \\ \boldsymbol{\omega}^2_{\max}\bar{\sigma}^2_{\omega^o}[(2\widehat{r}_z-\widehat{r}_x)^2+(2\widehat{r}_z-\widehat{r}_y)^2+(\widehat{r}_x+\widehat{r}_y)^2] \\ +\bar{\sigma}^4_{\omega^o}(4\widehat{r}_x^2+\widehat{r}_y^2+\widehat{r}_z^2)+\frac{65\bar{\boldsymbol{\sigma}}^2_{\omega}(\widehat{r}_y^2+\widehat{r}_z^2)}{72\Delta t^2}+\bar{\sigma}^2_{a^o} \\ +\bar{\sigma}^4_{\omega^o}(4\widehat{r}_y^2+\widehat{r}_x^2+\widehat{r}_z^2)+\frac{65\bar{\boldsymbol{\sigma}}^2_{\omega}(\widehat{r}_x^2+\widehat{r}_z^2)}{72\Delta t^2}+\bar{\sigma}^2_{a^o} \\ +\bar{\sigma}^4_{\omega^o}(4\widehat{r}_z^2+\widehat{r}_x^2+\widehat{r}_y^2)+\frac{65\bar{\boldsymbol{\sigma}}^2_{\omega}(\widehat{r}_x^2+\widehat{r}_y^2)}{72\Delta t^2}+\bar{\sigma}^2_{a^o} \end{bmatrix}
\end{aligned} \tag{29}$$

where $\boldsymbol{\omega}_{\max}$ is obtained by counting the weightless data, which is the maximum absolute value of the angular velocity of the three axes; $\widehat{r}_x$, $\widehat{r}_y$, and $\widehat{r}_z$ are the three components of the pre-estimated $\widehat{\boldsymbol{r}}$.

From (25) to (29), to summarize, the weight coefficients can be obtained by

$$\begin{cases} \boldsymbol{C}_1=\dfrac{1}{6\bar{\sigma}^4_{a^o}+4\boldsymbol{g}^2\bar{\sigma}^2_{a^o}} \\ \boldsymbol{C}_2=\dfrac{1}{\bar{\sigma}^2_{\omega^s}+3(\widehat{\boldsymbol{G}}_{\max})^2\bar{\sigma}^2_{a^o}} \\ \boldsymbol{C}_3=\dfrac{1}{2\bar{\sigma}^2_{a^o}+\boldsymbol{g}^2(4\sigma^2_{q^m}+8\sigma^4_{q^m}))} \\ \boldsymbol{C}_4=\dfrac{1}{\max(29)} \end{cases} \tag{30}$$

where $\widehat{\boldsymbol{G}}_{\max}$ is the largest absolute number in the matrix $\widehat{\boldsymbol{G}}$.

IV. Experiments

The proposed algorithm was validated and evaluated on both synthetic data and real-world data. In particular, the synthetic data was generated to simulate the sensors under different motions with artificial parameters. The parameters led by the proposed algorithm were then compared with the artificial parameters, and the variance of difference values between the two sets of parameters was presented and analyzed. The proposed algorithm has also been applied to real-world benchmark data set

for a comparative study in reference to the results obtained from the existing algorithms using the same IMU data. The comparative study was conducted using two measurement approaches, including the measurement of the accuracy of centripetal acceleration for accelerometer and the measurement of the accuracy of the attitude updating for the gyroscopes. In the third part of the experiments, the calculated parameters are applied to the real application scene, and a state detector is designed to detect whether the rigid body with an IMU device is in a falling state.

A. Experiment on Synthetic Data

The synthetic data contains two sets of data streams: RD1 is the sensor static-movement data and RD2 is the sensor data on rigid body in free fall with spinning. At first, noise-free data in the two sets is generated. Second, distort parameters of $\boldsymbol{T}, \boldsymbol{K}$, and $\boldsymbol{b}$ are added to transform the generated data. Third, the Gaussian noise $\boldsymbol{\nu}$ with 0 mean and σ_a for accelerometers and σ_g for gyroscopes is added. The following equations complete the conversion process for the accelerometers and gyroscopes:

$$\boldsymbol{a}^s_{\text{syn}} = \boldsymbol{f}_a^{-1}((\boldsymbol{T}^a)^{-1}\boldsymbol{a}^o_{\text{syn}}, \boldsymbol{K}^a) - \boldsymbol{b}^a + \boldsymbol{\nu}^a \tag{31}$$

$$\boldsymbol{\omega}^s_{\text{syn}} = \boldsymbol{f}_g^{-1}((\boldsymbol{T}^g)^{-1}\boldsymbol{\omega}^o_{\text{syn}}, \boldsymbol{K}^g) - \boldsymbol{b}^g - \boldsymbol{G}_g\boldsymbol{a}^o + \boldsymbol{\nu}^g. \tag{32}$$

In RD1 generation, $n_1 = 40$ different accelerometer vectors are randomly created under the condition that all norms of these vectors are $\|\boldsymbol{g}\|$. Expand each of accelerometer vectors with [0,0,0] gyroscope vector to be a sequence data in time $t_1 = 3$ s as stationary data. Calculate the angular velocity by dividing the angle between accelerometer vectors $\boldsymbol{a}^o_{i-1,\text{syn}}$ and $\boldsymbol{a}^o_{i,\text{syn}}$ in $t_2 = 2$ s. Meanwhile, the accelerometer vector area gradual linear transformation process from $\boldsymbol{a}^o_{i-1,\text{syn}}$ to $\boldsymbol{a}^o_{i,\text{syn}}$ in $t_2 = 2$ s. The time varying data accelerometer vector and angular velocity is the movement data. The stationary data and movement data are alternately connected.

In RD2 generation, $n_2 = 100$ set of accelerometer vectors and angular velocity $\boldsymbol{\omega}^o$ under weightlessness in $t = 1$ s is obtained by (33). This equation describes two combination movements: rotation and coning motion.

$$\begin{bmatrix} \omega^o_{x,\text{syn}} \\ \omega^o_{y,\text{syn}} \\ \omega^o_{z,\text{syn}} \end{bmatrix} = \frac{\boldsymbol{R}(\boldsymbol{q}_{\text{rand}})}{\sqrt{\boldsymbol{\omega}^2_{\text{set}} + 80^2}} \begin{bmatrix} \mathbf{sin}(\psi(t))\boldsymbol{\omega}_{\text{set}} \\ \mathbf{cos}(\psi(t))\boldsymbol{\omega}_{\text{set}} \\ 80 \end{bmatrix} \tag{33}$$

where $\boldsymbol{R}(\boldsymbol{q}_{\text{rand}})$ is a rotation matrix, which is obtained by substituting a randomly generated quaternion $\boldsymbol{q}_{\text{rand}}$ in (13). $\omega_{\text{set}} = 2\pi[\frac{1}{4}\text{rand}() + \frac{3}{4}]V_{\text{spin}}$ is used to randomly generate angular velocity vectors, where rand() is a function to obtain a random number with a mean of 0 and a variance of 1. V_{spin} is the velocity of the rigid body when it is rotating in the air; V_{spin} is set to 7. $\psi(t)) = 0.8t + 2\pi\text{rand}()$ is a function that generates a sequence of rotation axes. The angular acceleration $\dot{\boldsymbol{\omega}}_{\text{syn}}$ is derived from (33). The $\boldsymbol{a}^o_{\text{syn}}$ is obtained by substituting $\boldsymbol{\omega}_{\text{syn}}$ and $\dot{\boldsymbol{\omega}}_{\text{syn}}$ into (17) and (18). The sampling frequency of the entire synthetic data is fixed to 100 Hz.

Twenty sets of RD1 and RD2 are randomly generated using the above method so as to obtain 40 sets of parameters for the proposed algorithm. Then, the means and variances of each parameters are calculated for the linear model and nonlinear model. The experimental results are listed in Table I; for each calibration parameter, the three subtables in the upper part are: set values, calculated mean of the linear model, and calculated means of the nonlinear model; the two subtables of the lower part are: standard deviations of the linear model and standard deviations of the nonlinear model.

TABLE I
COMPARE PARAMETERS

Cross-Coupling-Accelerometer $\boldsymbol{T}^a$

1	0.03	-0.04	1	0.02963	-0.03952	1	0.03001	-0.04000
0	1	0.035	0	1	0.03523	0	1	0.03502
0	0	1	0	0	1	0	0	1
			0	2.2e-6	1.4e-6	0	2.4e-8	2.1e-8
			0	0	1.6e-6	0	0	1.5e-8
			0	0	0	0	0	0

Cross-Coupling-Gyroscope $\boldsymbol{T}^g$

1	-0.02	0.04	1	-0.02023	0.03979	1	-0.02002	0.04002
0.02	1	-0.02	0.02027	1	-0.01980	0.01998	1	-0.01999
-0.026	0.03	1	-0.02705	0.03014	1	-0.02606	0.02995	1
			0	8.5e-6	5.1e-6	0	3.1e-8	3.1e-8
			7.0e-6	0	6.8e-6	2.6e-8	0	2.6e-8
			7.4e-6	9.6e-6	0	4.2e-8	2.5e-8	0

Scale Factor-Accelerometer $\boldsymbol{K}^a$

9.81	-0.03	0.04	9.83011	0	0	9.80983	-0.02998	0.03996
9.79	0.04	0.04	9.8108	0	0	9.78998	0.03971	0.03998
9.76	0.03	0.06	9.7962	0	0	9.76000	0.03003	0.05992
			5.2e-5	0	0	7.6e-7	3.1e-6	6.8e-8
			5.4e-5	0	0	1.0e-6	4.6e-6	6.4e-8
			3.6e-5	0	0	9.3e-7	4.6e-6	1.4e-7

Scale Factor-Gyroscope $\boldsymbol{K}^g$

3.12	0.015	0.023	3.12091	0	0	3.12003	0.01498	0.02300
3.15	-0.027	0.021	3.15932	0	0	3.14993	-0.02701	0.02101
3.09	0.02	0.03	3.09132	0	0	3.08993	0.01997	0.03000
			5.7e-5	0	0	1.7e-7	2.4e-8	2.4e-9
			3.6e-5	0	0	1.9e-7	2.1e-8	2.1e-9
			7.1e-5	0	0	1.7e-7	9.2e-8	3.8e-9

Bias-Accelerometer $\boldsymbol{b}^a$

0.04	0.5	-0.35	0.03850	0.50216	-0.34845	0.03999	0.50004	-0.35000
			2.6e-7	3.9e-7	2.0e-7	1.3e-8	1.5e-8	2.2e-8

Bias-Gyroscope $\boldsymbol{b}^g$

0.04	-0.15	0.55	0.40006	-0.15014	0.55010	0.40000	-0.15000	0.55000
			1.8e-8	1.6e-8	1.4e-8	1.2e-10	1.7e-10	1.4e-10

g-dependent Bias Matrix $\boldsymbol{G}^g(10^{-4})$

1	2	-1	0	0	0	1.00290	1.99882	-1.00738
-1	2	1	0	0	0	-1.00012	1.99770	1.00605
2	3	1	0	0	0	1.99540	3.00467	1.00001
			0	0	0	8.8e-12	5.9e-12	7.0e-12
			0	0	0	5.5e-12	7.2e-12	6.9e-12
			0	0	0	8.0e-12	6.1e-12	9.6e-12

Radius $\boldsymbol{r}$

0.07	0.07	0.07	0.09595	0.09530	0.09034	0.06998	0.06997	0.06997
			1.4e-6	1.8e-6	1.2e-6	1.3e-8	1.3e-8	1.3e-8

The calculation results show that under the conditions of the data generated by the nonlinear model, the calculation results of the nonlinear parameters are closer to the true values than those of the linear model; in addition, the variances of our method are also smaller than those of the linear model.

B. Experiment on Real-World Data

So far, IMU calibration does not have a consensus public data set. In this article, we try to use the most simple tools without the dependency of the accuracy of the tools to obtain more

Fig. 4. Hollow steel ball and IMU sensor. The IMU sensor is fixed on the hollow steel ball.

accurate calibration parameters through calibrations. However, we can find only one method, i.e., the work of Tedaldi *et al.* [6], which does not depend on the accuracy of tools. Moreover, the work of Tedaldi *et al.* work has the most citations on the IMU calibration during recent years. Other methods must rely on precise equipment, such as [20] and [21]. Several methods must use instruments designed by themselves [22], [23]. It is extremely difficult for us to reproduce these devices. Hence, our method is only compared with the work of Tedaldi *et al.* work in real-world data.

In this experiment, the IMU sensor ICM42605 from Invensense was used for calibration. The full scale range of ICM42605 is $\pm$2000 dps (degrees per second) for gyroscope and $\pm$16 g for accelerometer. The parameters are set as follows: for RD1, $M = 20$, $T_{\text{init}} = 60$ s, $T_w = 2$ s, and $n = 6$; for RD2, $N = 100$, $t_w = 0.05$, and $n = 1000$. A hollow steel ball was used as the rigid body in the experiment. The IMU sensor was mounted on the surface of the ball, as shown in Fig. 4. To collect RD1, the ball was manually switched to a different posture for a period t_w. To collect RD2, the ball was thrown vertically and rotated, the rotation axis is random, and 200 sets of data were collected under such conditions.

The accuracy of the algorithm was accessed by an indirect method to avoid the use of a professional measuring instrument. The test data RD3 was collected by throwing the hollow steel ball vertically, catching the ball and placing it on the stable base; this was repeated for ten times and then ten data sets were collected. The accuracy of the acceleration calibration was determined by calculating the difference between the acceleration measurement and the centripetal force and the Euler force. The accuracy of the gyroscope calibration was accessed by examining the angle between the gravity calculated by the attitude and the true gravity. Note that the more accurate the gyroscope is calibrated, the smaller the angle becomes.

The acquisition of RD1 is also applicable to the calibration algorithm of *Tedaldi D* [6], which is a linear model. In this experiment, a set of calibration parameters is calculated by the algorithm *Tedaldi D*. Four combinations of the parameters, including $Linear$, $Quadratic$, $Cube$, and All are calculated using the proposed algorithm. Here, $Linear$ indicates that the scale factors of the calibration model is linear and the g-dependent bias $\boldsymbol{G}^g$ is ignored; and in the calculation process, only RD1 is used and $\boldsymbol{C}_4$ in the global cost function is set to 0; $Quadratic$ and $Cube$, respectively, indicate that the scale factors of the calibration model are quadratic and cube, and the g-dependent bias $\boldsymbol{G}^g$ is ignored. All indicates that all parameters mentioned in this article are calculated. The calculated parameters that are obtained by *Tedaldi D* and the proposed algorithms are summarized in Table II.

TABLE II
CALCULATED PARAMETERS

	Tedaldi D	Our Method			
		linear	quadratic	cube	all
α_{yz}	1.4387e-04	1.5988e-03	1.3055e-03	1.4718e-03	1.4910e-03
α_{zy}	-1.6944e-03	-3.8875e-03	-3.3402e-03	-3.3564e-03	-3.3904e-03
α_{zx}	4.5197e-04	3.5809e-03	3.7436e-03	3.6216e-03	3.6206e-03
$b_x^a(m/s^2)$	1.5359e-03	5.6602e-04	1.1061e-03	9.7759e-04	9.9456e-04
$b_y^a(m/s^2)$	1.8357e-03	2.8507e-03	3.0009e-03	3.0966e-03	3.0840e-03
$b_z^a(m/s^2)$	-3.575e-03	-3.8443e-03	-4.0922e-03	-3.8634e-03	-3.8754e-03
γ_{yz}	-4.4878e-03	-5.2886e-04	-5.998e-04	-5.2218e-04	-4.9946e-04
γ_{zy}	-2.3963e-03	-1.0262e-02	-1.0011e-02	-1.0019e-02	-1.0046e-02
γ_{xz}	7.1768e-03	5.8778e-03	5.8353e-03	5.8536e-03	5.8725e-03
γ_{zx}	4.1875e-03	2.1119e-03	2.3471e-03	2.276e-03	2.2758e-03
γ_{xy}	1.1272e-02	9.1793e-03	1.0216e-02	1.02e-02	1.0138e-02
γ_{yx}	4.0622e-04	2.652e-03	2.8927e-03	2.8124e-03	2.8249e-03
$b_x^g(rad/s)$	-1.2961e-03	-1.4304e-03	-1.3405e-03	-1.3435e-03	-1.3494e-03
$b_x^g(rad/s)$	-1.2349e-03	-1.6229e-03	-1.3244e-03	-1.3271e-03	-1.3124e-03
$b_z^g(rad/s)$	1.9384e-05	1.0797e-04	4.5239e-05	4.8474e-05	6.8465e-05
$r_x(m)$		1.4096e-03	1.4057e-03	1.4045e-03	1.4082e-03
$r_y(m)$		5.8362e-03	5.852e-03	5.8415e-03	5.8413e-03
$r_z(m)$		-1.0941e-01	-1.0942e-01	-1.0937e-01	-1.0938e-01
$k_{11}^a(m/s^2)$	9.7882e+00	9.7958e+00	9.7935e+00	9.7976e+00	9.7976e+00
k_{12}^a			-9.7025e-03	-9.3689e-03	-9.3256e-03
k_{13}^a				-6.7377e-04	-7.2719e-04
$k_{21}^a(m/s^2)$	9.7926e+00	9.7964e+00	9.7945e+00	9.7976e+00	9.7974e+00
k_{22}^a			-6.8205e-04	7.2986e-04	7.9315e-04
k_{23}^a				-7.5889e-04	-7.7014e-04
$k_{31}^a(m/s^2)$	9.7848e+00	9.7855e+00	9.7885e+00	9.7856e+00	9.7857e+00
k_{32}^a			1.8747e-03	2.3233e-05	1.1308e-04
k_{33}^a				-1.5299e-04	-1.4873e-04
$k_{11}^g(rad/s)$	3.1351e+00	3.1323e+00	3.1327e+00	3.1346e+00	3.1345e+00
k_{12}^g			-3.1524e-05	-3.6341e-05	-2.3754e-05
k_{13}^g				-1.6504e-05	-1.5991e-05
$k_{21}^g(rad/s)$	3.1125e+00	3.1183e+00	3.119e+00	3.1181e+00	3.1181e+00
k_{22}^g			-3.955e-05	-2.1107e-05	-5.1553e-05
k_{23}^g				2.7426e-05	2.6358e-05
$k_{31}^g(rad/s)$	3.1228e+00	3.1196e+00	3.1222e+00	3.1183e+00	3.1183e+00
k_{32}^g			-2.9493e-04	-2.3602e-04	-2.3978e-04
k_{33}^g				1.1842e-04	1.1732e-04
$G_1(rad*s/m)$					-3.4576e-06
$G_2(rad*s/m)$					-3.1213e-06
$G_3(rad*s/m)$					2.1329e-06
$G_4(rad*s/m)$					-8.7706e-06
$G_5(rad*s/m)$					-3.5515e-07
$G_6(rad*s/m)$					-7.1096e-06
$G_7(rad*s/m)$					-1.6885e-05
$G_8(rad*s/m)$					-8.1315e-07
$G_9(rad*s/m)$					-2.4591e-06

The accuracies of the acceleration calibration from different sets of parameters as listed in Table II are summarized in Table III. In particular, the RD3 is extracted by the static detector operator, and the error is calculated by

$$\boldsymbol{L}(\boldsymbol{\theta}^a, \boldsymbol{\theta}^g, \boldsymbol{r}) = \sum_{j=1}^{S} \|\boldsymbol{a}_j^o - \boldsymbol{f}_C(\boldsymbol{\omega}_j^o, \boldsymbol{r}) - \boldsymbol{f}_E(\dot{\boldsymbol{\omega}}_j^o, \boldsymbol{r})\|^2. \quad (34)$$

Note that a smaller error indicates better accuracy of the calibration parameters. S represents the number of frames of RD3 in a weightless state.

The accuracies of the gyroscopes calibration from different sets of parameters are listed in Table IV. A performance indicator is designed as the cos value of the angle between the two vectors

TABLE III
PARAMETERS USED TO COMPARE ACCELEROMETERS

Tedaldi D	Our Method			
	linear	quadratic	cube	all
66.31	15.96	14.37	13.70	**13.62**
43.68	20.60	17.27	13.31	**12.24**
34.88	17.35	15.63	15.92	**15.12**
151.10	89.81	88.65	89.34	**87.50**
114.52	45.41	32.88	**30.12**	30.37
123.58	48.95	45.54	37.02	**36.72**
197.54	60.41	58.26	**52.61**	52.28
362.41	230.00	204.44	**191.81**	192.14
201.92	68.36	59.41	**56.67**	56.86
342.84	177.00	140.16	**131.85**	132.35

TABLE IV
PARAMETERS USED TO COMPARE GYROSCOPES

Tedaldi D	Our Method				angle(π)
	linear	quadratic	cube	all	
0.99962	0.99813	0.99924	0.99913	0.99932	30.6
0.96867	0.95936	0.97081	0.98862	**0.98881**	55.9
0.87617	0.84341	0.88079	**0.95506**	0.95282	77.7
0.94326	0.95452	0.98280	0.99450	**0.99546**	99.8
0.95350	0.95584	0.98621	0.98905	**0.99092**	115.6
0.95631	0.95619	**0.97603**	0.94215	0.94735	128.7
0.60969	0.67034	0.79010	**0.94798**	0.94109	140.2
0.68052	0.73106	0.85015	**0.97835**	0.97443	148.0
0.71950	0.72965	0.84653	**0.98860**	0.98521	154.8
0.72404	0.79269	0.90238	**0.99922**	0.99828	159.0

$\boldsymbol{a}_j^o$ and $\boldsymbol{R}_{j-1,j}\boldsymbol{a}_{j-1}^o$

$$\boldsymbol{L}(\boldsymbol{\theta}^a, \boldsymbol{\theta}^g) = \sum_{j=2}^{S+1} \frac{\boldsymbol{a}_j^{oT} \boldsymbol{R}_{j-1,j} \boldsymbol{a}_{j-1}^o}{\|\boldsymbol{a}_j^o\| \|\boldsymbol{a}_{j-1}^o\|}. \tag{35}$$

The closer the cos value to 1, the more accurate the calibration parameters are. The parameter r was not used in the models in *Tedaldi D* and *Linear* in the process of calculating the performance indicator, but r was used when calculating *All*.

The ranking of the precisions from low to high of different parameters are summarized in Tables III and IV, including *Tedaldi D*, *Linear*, *Quadratic*, *Cube*, and *All*. In particular, *Linear* and *Tedaldi D* shared identical data and model parameters, but *Linear* has achieved relatively better performances for the global cost function for the overall optimization of the parameters. *Quadratic* has shown a better performance than *Linear* for the accelerator and angular velocity. Compared with *Linear* and *Quadratic*, *Cube* had an essential improvement in the acceleration and angular velocity. *All* has one more parameter, g-dependent bias $\boldsymbol{G}^g$, than *Cube*. The average reduction percentage of the accelerometers error is 64.12% in Table III, obtained by: $\sum(1 - \boldsymbol{L}(\boldsymbol{\theta}_{\text{all}}^a, \boldsymbol{\theta}_{\text{all}}^g, \boldsymbol{r}_{\text{all}})/\boldsymbol{L}(\boldsymbol{\theta}_{\text{Tedaldi}}^a, \boldsymbol{\theta}_{\text{Tedaldi}}^g, \boldsymbol{r}_{\text{all}}))/N$. Also, the average reduction percentage of the gyroscopes error is 47.90% in Table IV, obtained by: $\sum(1 - \arccos(\boldsymbol{L}(\boldsymbol{\theta}_{\text{all}}^a, \boldsymbol{\theta}_{\text{all}}^g))/\arccos(\boldsymbol{L}(\boldsymbol{\theta}_{\text{Tedaldi}}^a, \boldsymbol{\theta}_{\text{Tedaldi}}^g)))/N$.

By observing Tables III and IV, the g-dependent bias $\boldsymbol{G}^g$ does not substantially improve the accuracy of the IMU calibration, compared to *Cube*. The reason is that the accuracy of the MEMS sensor used in this experiment cannot accurately calculate the true g-dependent bias $\boldsymbol{G}^g$. To summarize, the nonlinear scale factor has led to better performance over a larger range of the sensor, and the global cost function further improved the accuracy.

TABLE V
PARAMETERS OF IMU IN IPHONE

$\boldsymbol{T}^a$

1	-1.9609e-03	5.3115e-03
0	1	1.4379e-03
0	0	1

$\boldsymbol{T}^g$

1	3.1193e-04	6.8747e-03
2.2605e-03	1	8.7853e-04
2.6503e-03	-2.4195e-03	1

$\boldsymbol{K}^a$

1.0018	-2.0349e-04	-2.9115e-05
1.0015	-5.3635e-05	-2.2143e-05
0.99638	-1.2255e-04	5.1980e-05

$\boldsymbol{K}^g$

0.99827	6.2259e-05	-7.4481e-06
1.0070	1.6163e-04	-2.9095e-05
0.99831	-8.1644e-04	8.3276e-06

$\boldsymbol{B}^a(m/s^2)$

-7.6651e-03	2.0240e-02	9.0172e-02

$\boldsymbol{B}^g(rad/s)$

-9.4971e-05	7.2700e-05	1.7860e-04

$\boldsymbol{r}(m)$

-1.4174e-02	-2.0610e-02	1.9555e-03

$\boldsymbol{G}^g$

-8.7428e-07	-1.8119e-05	-2.5090e-06
8.6587e-07	-1.4582e-05	-1.1737e-05
-1.8528e-05	2.7332e-05	1.8225e-05

The models and algorithms presented in this article outperformed those referenced approaches utilized in the experiments. In the calibration model, the scale factor of the sensor conforms to a linear property in a small range, that is, a fixed value, but the scale factor varies if a larger range of the sensor is used, which led to the necessity of a nonlinear scale factor. This is achieved by globally optimizing the cost function such that all parameters are optimized in the same time. In addition, the weights of the subcost functions also ensure that the calibrated parameters are achieved for an optimal solution.

C. Experiment on Application Scene

In the experiment, a state detector is designed to detect whether a rigid body (a mobile phone) with an IMU device is in the falling state. The rigid body in the experiment is chosen as "iphone 8," and the "Phyposx App" in the iphone is utilized to collect the IMU data.

Using the algorithm mentioned above, we calculate the parameters of the IMU. In the collection of RD1, the iphone was placed in different attitudes for 20 times. In the collection of RD2, the iphone was thrown up by different rotation axes and different rotation speeds. The iphone was thrown for 12 times in the experiment. Note that a soft bed surface was chosen for the throwing place to avoid breaking the iphone.

The parameters present in Table V are the processed results by our IMU calibration method. Our method successfully generated the parameters for $\boldsymbol{T}^a$, $\boldsymbol{T}^g$, $\boldsymbol{K}^a$, $\boldsymbol{K}^g$, $\boldsymbol{B}^a$, $\boldsymbol{B}^g$, $\boldsymbol{G}^g$, and $\boldsymbol{r}$. The position of the iphone's gravity center in the IMU coordinate system is $[-14.174, -20.610, 1.9555]$ mm.

Fig. 5 illustrates the position of IMU in the phone's coordinate system. The origin is the gravity center, assuming that the gravity center of the iphone is in the device's physical center.

The IMU data and parameters calculated by our algorithm are substituted into (19) to obtain the loss. A threshold was set at 2.5 m/s^2. If the loss is less than the threshold, the rigid body is judged as in a falling state (if an object is static, the result of (19) should be 9.8 m/s^2).

Fig. 6 illustrates the falling state detector results of the IMU data, which was collected when the iphone was thrown up for other eight times. The detector's data is shown on the top and IMU raw data is on the bottom of the figure. At the top of Fig. 6,

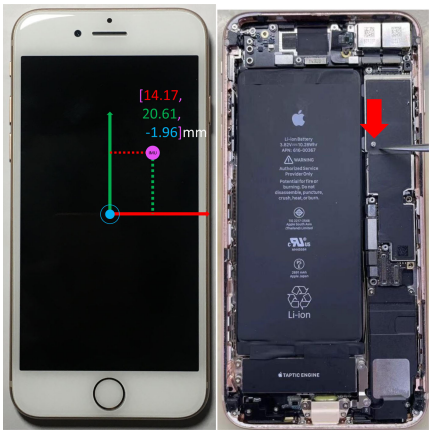

Fig. 5. Position of IMU in the phone's coordinate system. On the left side, the pink point is the calculated IMU position. On the right side, the position pointed by the red arrow is the real position of the IMU in an iphone 8.

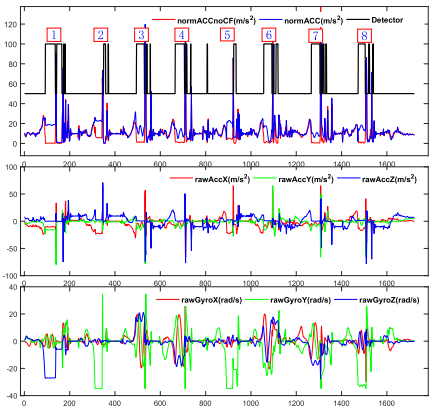

Fig. 6. Falling state detector and the raw data of IMU. The upper part is the falling state detector. The middle and bottom parts are the raw data of accelerometers and gyroscopes, respectively.

the red line is the acceleration of the resultant force that removes centripetal force and Euler force. The blue line is the acceleration that comes from normalizing the raw data of accelerometers. The black line is a falling state detector. If the line is on the value 100, this situation indicates that the iphone is in a falling state. This figure illustrates that our method can successfully detect almost all the falling states. However, in Fig. 6, the second and fifth falling states were not detected. The reason is that the original angular velocity data of the *Y*-axis of the gyroscope went beyond the measurement range of the iphone, i.e., the detector device has not obtained the real angular velocity data; thus, the judgment fails.

V. Conclusion

This article presented a method for the calculation of the parameters of IMUs when they are used in a large range, in which the scale factors are nonlinear. By placing the IMU in a rigid object in free fall and rotation states, a large amount of data from the accelerometer and gyroscopes can be obtained, which provides the necessary data for calculating nonlinear scale factors of the IMU. In the calibration process, the global cost function was used, and each of the subcost functions was weighted to ensure the global optimum. The experiments on the synthesis and real-world data sets showed that the proposed model and algorithm are feasible; the performance of this method is better than that of *Tedaldi*'s method: The errors of acceleration and angular velocity led by our algorithm are significantly smaller than those resulted from the existing approaches, demonstrating a remarkable improvement of 64.12% and 47.90%, respectively. In the application scene, a falling state detector with the proposed IMU calibration was designed and demonstrated our approach's effectiveness.

There is still room for improving this work. For example, the calibrated IMU can be integrated with a camera to create a brush writing motion capturing system. Thus, a data set including the handwriting trajectories of the Chinese characters or English letter can be conveniently obtained.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive comments, which helped greatly in revising this work.

References

[1] B. Wang, Z. Deng, C. Liu, Y. Xia, and M. Fu, "Estimation of information sharing error by dynamic deformation between inertial navigation systems," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 2015–2023, Apr. 2014.

[2] S. Jiang *et al.*, "Feasibility of wrist-worn, real-time hand, and surface gesture recognition via SEMG and IMU sensing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3376–3385, Aug. 2018.

[3] F. Lin, A. Wang, Y. Zhuang, C. Zhang, M. R. Tomita, and W. Xu, "Smart insole: A wearable sensor device for unobtrusive gait monitoring in daily life," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2281–2291, Dec. 2016.

[4] X. Tong *et al.*, "A double-step unscented Kalman filter and HMM-based zero velocity update for pedestrian dead reckoning using MEMS sensors," *IEEE Trans. Ind. Electron.*, vol. 67, no. 1, pp. 581–591, Jan. 2020.

[5] S. Poddar, V. Kumar, and A. Kumar, "A comprehensive overview of inertial sensor calibration techniques," *J. Dyn. Syst., Meas., Control*, vol. 139, no. 1, Jan. 2017, Art. no. 011006.

[6] D. Tedaldi, A. Pretto, and E. Menegatti, "A robust and easy to implement method for IMU calibration without external equipments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3042–3049.

[7] P. Gao, K. Li, T. Song, and Z. Liu, "An accelerometers-size-effect self-calibration method for triaxis rotational inertial navigation system," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1655–1664, Feb. 2018.

[8] B. Wang, Q. Ren, Z. Deng, and M. Fu, "A self-calibration method for nonorthogonal angles between gimbals of rotational inertial navigation system," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2353–2362, Apr. 2015.

[9] Q. Cai, N. Song, G. Yang, and Y. Liu, "Accelerometer calibration with nonlinear scale factor based on multi-position observation," *Meas. Sci. Technol.*, vol. 24, no. 10, Aug. 2013, Art. no. 105002.

[10] H. Sohrabi and S. Ebadollahi, "Accuracy enhancement of MEMS accelerometer by determining its nonlinear coefficients using centrifuge test," *Measurement*, vol. 112, pp. 29–37, Dec. 2017.

[11] A. Olivares, G. Olivares, J. M. Gorriz, and J. Ramirez, "High-efficiency low-cost accelerometer-aided gyroscope calibration," in *Proc. IEEE Int. Conf. Test Meas.*, 2009, pp. 354–360.

[12] A. Kozlov, I. Sazonov, and N. Vavilova, "IMU calibration on a low grade turntable: Embedded estimation of the instrument displacement from the axis of rotation," in *Proc. IEEE Int. Symp. Inertial Sensors Syst.*, 2014, pp. 1–4.

[13] E. E. Aktakka, J.-K. Woo, and K. Najafi, "On-chip characterization of scale-factor of a MEMS gyroscope via a micro calibration platform," in *Proc. IEEE Int. Symp. Inertial Sensors Syst.*, 2017, pp. 1–4.

[14] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Norwood, MA, USA: Artech House, 2013.
[15] C. Jekeli, *Inertial Navigation Systems With Geodetic Applications*. Berlin, Germany: Walter de Gruyter, Oct. 2012.
[16] P. Schopp, H. Graf, W. Burgard, and Y. Manoli, "Self-calibration of accelerometer arrays," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 8, pp. 1913–1925, Aug. 2016.
[17] N. El-Sheimy, H. Hou, and X. Niu, "Analysis and modeling of inertial sensors using allan variance," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 1, pp. 140–149, Jan. 2008.
[18] Y. F. Jiang and Y. P. Lin, "Improved strapdown coning algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 2, pp. 484–490, Apr. 1992.
[19] P. G. Savage, "Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms," *J. Guid., Control, Dyn.*, vol. 21, no. 1, pp. 19–28, Jan. 1998.
[20] P. Schopp, H. Graf, W. Burgard, and Y. Manoli, "Self-calibration of accelerometer arrays," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 8, pp. 1913–1925, Aug. 2016.
[21] U. Qureshi and F. Golnaraghi, "An algorithm for the in-field calibration of a MEMS IMU," *IEEE Sensors J.*, vol. 17, no. 22, pp. 7479–7486, Nov. 2017.
[22] H. Sohrabi and S. Ebadollahi, "Accuracy enhancement of MEMS accelerometer by determining its nonlinear coefficients using centrifuge test," *Measurement*, vol. 112, pp. 29–37, Dec. 2017.
[23] H. Wei, M. Wu, and J. Cao, "New matching method for accelerometers in gravity gradiometer," *Sensors*, vol. 17, no. 8, Aug. 2017, Art. no. 1710.

Xin Zhang received the B.Sc. degree in mathematics and applied mathematics from the School of Mathematical Sciences, Xiamen University, Xiamen, China, in 2009, and the M.Sc. degree in intelligence science and technology from the School of Informatics, Xiamen University, in 2014. He is currently working toward the Ph.D. degree in intelligence science and technology with the School of Informatics, Xiamen University.

His research interests include inertial sensors, calibration, data fusion, motion capture, and pattern recognition.

Changle Zhou received the Ph.D. degree in computer science from Peking University, Beijing, China, in 1990.

He is currently a Professor with the Artificial Intelligence Department, Xiamen University, Xiamen, China. He is also an affiliated Professor of linguistics and applied linguistics with the Humanity College, Zhejiang University, Hangzhou, China, and an affiliated Professor with the Philosophy Department, Xiamen University. His research interests lie in the areas of artificial intelligence. His scientific contribution to the AI has more to do with machine consciousness and the logic of mental self-reflection. Beyond AI project, he also carries out research on a host of other topics including computational brain modeling, computational modeling of analogy and metaphor and creativity, computational musicology, and information processing of data regarding traditional Chinese medicine. His philosophical works lie in ancient oriental thoughts of Chinese, such as ZEN, TAO, YI, etc., viewed from science.

Fei Chao (Member, IEEE) received the B.Sc. degree in mechanical engineering from the Fuzhou University, Fuzhou, China, in 2004, the M.Sc. degree with distinction in computer science from the University of Wales, Cardiff, U.K., in 2005, and the Ph.D. degree in robotics from the Aberystwyth University, Aberystwyth, U.K., in 2009.

He is currently an Associate Professor with the School of Informatics, Xiamen University, Xiamen, China. He has authored or coauthored more than 100 peer-reviewed journal and conference papers. His research interests include developmental robotics, machine learning, and optimization algorithms.

Chih-Min Lin (Fellow, IEEE) was born in Taiwan, in 1959. He received the B.S. and M.S. degrees in control engineering from the Department of Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree in electronics engineering from the Institute of Electronics Engineering, National Chiao Tung University, in 1986.

He is currently a Chair Professor and the Vice President with Yuan Ze University, Chung-Li, Taiwan. His current research interests include fuzzy neural network, cerebellar model articulation controller, intelligent control systems, and signal processing.

Dr. Lin also serves as an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS and IEEE TRANSACTIONS ON FUZZY SYSTEMS.

Longzhi Yang (Senior Member, IEEE) received the B.Sc. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 2003, the M.Sc. degree in computer science from Coventry University, Coventry, U.K., in 2006, and the Ph.D. degree in computer science from Aberystwyth University, Aberystwyth, U.K., in 2011. He is currently a Programme Leader and a Reader with Northumbria University, Newcastle upon Tyne, U.K. He is the Founding Chair of the IEEE Special Interest Group on Big Data for Cyber Security and Privacy. His research interests include computational intelligence, machine learning, big data, computer vision, intelligent control systems, and the application of such techniques in real-world uncertain environments.

Dr. Yang is the recipient of the Best Student Paper Award of the 2010 IEEE International Conference on Fuzzy Systems.

Changjing Shang received the Ph.D. degree in computing and electrical engineering from Heriot-Watt University, Edinburgh, U.K., in 1995.

She is a University Research Fellow with the Department of Computer Science, Institute of Mathematics, Physics, and Computer Science, Aberystwyth University, Aberystwyth, U.K. Prior to joining Aberystwyth, she worked with Heriot-Watt University; Loughborough University, Loughborough, U.K.; and Glasgow University, Glasgow, U.K. Her research interests include pattern recognition, data mining and analysis, space robotics, and image modeling and classification.

Qiang Shen received the Ph.D. degree in computing and electrical engineering from Heriot-Watt University, Edinburgh, U.K., in 1990, and the D.Sc. degree in computational intelligence from Aberystwyth University, Aberystwyth, U.K., in 2013.

He holds the Established Chair in Computer Science and is the Pro Vice-Chancellor with the Faculty of Business and Physical Sciences, Aberystwyth University. He has authored two research monographs and more than 400 peer-reviewed papers. His research interests include computational intelligence and its application in robotics.

Dr. Shen is a recipient of the Outstanding Transactions Paper Award from the IEEE for one of his peer-reviewed papers.